Interpretation of Wangchen codes

‘W:\data\scripts\Software’ tetTool copied from ‘X:\ephy\_wangchen\wangchen\_Cdrive\_data\_code\Work\Experiment\Program’

There are several functions useful

**Depth analysis**

**1. ttshow.m**

This gui software basically work for loading excel file (e.g., Acute\_Feb15\_2012.xls) and presenting the depth of each tetrode

And it can export a mat file (e.g., Acute\_Feb15\_2012\_ttshow\_tetrodeDepth.mat).

In this mat file,

There is a structure array variable “tet” (n=1,…, 28; 24 tetrodes and 4 references)

Each structure component has

id: 'tt1'

depth: 1668

turns: [18x1 double]

turnStop: [18x1 double]

totalTurns: 34.8500

inBrain: 14

turnDepth: [18x1 double]

ref: 14

scale: 80

**2. some restructuring data across the 3 experiments and call makeTetrodeDepthDatabase.m**

Since Wangchen did not create a script file, I created based on the work flow of his work.

This script file is called ‘**wrapper\_makeTetrodeDepthDatabase.m**’

The final depth database is saved in

**‘W:\data\Wangchen\Acute Experiment Excel Log\tetrode\_depth\_redundencyCorrected.mat’**

3. getTetrodeDepth.m

e.g.,

load('W:\data\Wangchen\Acute Experiment Excel Log\tetrode\_depth\_redundencyCorrected.mat')

D = getTetrodeDepth(tetDepth,1,'FlashingBar',1:24)

**Spike Identification and sorting process (alex\_sorting)**

Spike detection –wangchen modified ‘alex\_sorting’ and create scripts

1. rtDetection.m

2. batchDetection.m

3. detectSpikesTetrodes.m

This code from wangchen directory

e.g.,

i=1

fn='W:\data\test\CEREBUS\DataFile\CerebusData\acute\NormLuminance\2011-Oct-19\04-13-01\acute\_NormLuminance005'

tetrode=1

detectSpikesTetrodes(sprintf('%s%s',fn,'.\*'),tetrode(i),sprintf('Sc%d.Htt',tetrode(i)));

Basically, **detectSpikesTetrodes.m is commensurate with detectSpikesTetrodesV2.m**

After spike detection, the data were save in ‘\*.Htt’

This file is loaded with e.g.,

fn='W:\data\test\CEREBUS\DataFile\CerebusData\acute\NormLuminance\2011-Oct-19\04-13-01\Sc1.Htt'

tt = ah\_readTetData(fn,'all'); %.Htt file

tt =

t: [65362x1 double]

w: {[28x65362 double] [28x65362 double] [28x65362 double] [28x65362 double]}

h: [65362x4 double]

tstart: []

tend: []

aligned: 1

t: time stamp for each spike in **millisecond**

w: waveforms (28 time sample points) from all identified spikes

h: height

**SpikeSorting – procedures**

mainSorting.m --- automatic spike sorting – generate modelX.mat

automan.m (from /tetTool)

fd='/media/sdd\_HGST6T/data/test/CEREBUS/DataFile/CerebusData/acute/NormLuminance'

automan(fd,0,1,0) --- automatic manual clusters generation.

**Stimulus event identification**

Load photodiode signal from the raw file

**1. saveTimestampsToMat.m** save stimDat.mat file

stimData.Values = stimValues; %stimulus intensity values (gray values centered and normalized to mean value 128).

stimData.Timestamps = t\_SETS; %timestamps of each stimlus values

stimData.Onsets = stimOnsets; %timestamps of each contrast cycle.(odd for low, even for high contrast)

**Timestamps are obtained from NEV file. Unit(Second)**

**2. syncPTBPTD(stimData**)

% synchronize the timestamps from psychtoolbox to photodiode signal

% synchPTBPTD returns the interpolated photodiode timestamps in the same length as that from psychtoolbox

%

stimData.ptbSyncTimes = swapTimes;

This function save ptbSyncTimes into stimData.

**syncPTBPTDData.m** is a batch script to call synchPTBPTD from a root directory.

3. extractPTDRaw

nsxfile = 'f:\Cerebus\DataFile\CerebusData\acute\FlashingBar\2011-Oct-20\00-24-19\acute\_FlashingBar003.ns5'

extractPTDRaw(nsxfile)

%extract the continuous ptd signal from nsx file

3. checkReg.m 🡪 validate the timing consistency of visual stimulation